

## 实验五 综合案例（3）

### 一、 实验目的

1、 掌握指针的定义与操作

2、 掌握字符串的操作

### 二、 实验内容

1、 PTA 作业《C++语言程序设计工程实践：内存编程》所有内容

2、 使用 Visual Studio 环境来编辑、编译和运行下列程序。

(1)用一个整数来表示银行账号,但这并不是完美的方案。例如,如果银行账号以“0”开头,或账号超过整数的表示范围,或账号中包括其他字符,这种表示方式都不能胜任。本章学习了字符串后,可以改用字符串来表示银行账号,这样以上问题得到了解决。另外,程序中所输出的账目列表,每笔账目都没有说明,使用字符串可以为各笔账目增加说明文字。此外,我们为 SavingsAccount 类专门增加了一个用来报告错误的函数,当其他函数需要输出错误信息时,直接把信息以字符串形式传递给该函数即可,简化了错误信息的输出。

(2)主程序创建的两个账户为两个独立的变量,只能用名字去引用它们,在主程序末尾分别对两个账户进行结算(settle)和显示(show)时,需要将几乎相同的代码书写两遍,如果账户数量增多将会带来更大麻烦。本章学习了数组后,可以将多个账户组织在一个数组中,这样可以把需要对各个账户做的事情放在循环中,避免了代码的冗余。

(3)日期都是用一个整数来表示的,这样计算两个日期相距天数时非常方便,但这种表示很不直观,对用户很不友好。事实上,日期可以用一个类来表示,内含年、月、日三个数据成员,但这又给计算两个日期相差天数带来了麻烦。为了计算日期间相差的天数,可以先选取一个比较规整的基准日期,在构造日期对象时将该日期到这个基准日期的相对天数计算出来,我们将这个相对天数称为“相对日期”。这样在计算两个日期相差的天数时,只需将二者的相对日期相减即可。

//代码内容 1：指针和字符串的使用

```
//date.h
#ifndef __DATE_H__
#define __DATE_H__
class Date { //日期类
private:
    int year;    //年
    int month;   //月
    int day;     //日
```

```

        int totalDays;    //该日期是从公元元年 1 月 1 日开始的第几天
public:
    Date(int year, int month, int day); //用年、月、日构造日期
    int getYear() const { return year; }
    int getMonth() const { return month; }
    int getDay() const { return day; }
    int getMaxDay() const;           //获得当月有多少天
    bool isLeapYear() const { //判断当年是否为闰年
        return year % 4 == 0 && year % 100 != 0 || year % 400 == 0;
    }
    void show() const;               //输出当前日期
    //计算两个日期之间差多少天
    int distance(const Date& date) const {
        return totalDays - date.totalDays;
    }
};
#endif //__DATE_H__

//date.cpp
#include "date.h"
#include <iostream>
#include <cstdlib>
using namespace std;
namespace { //namespace 使下面的定义只在当前文件中有效
    //存储平年中的某个月 1 日之前有多少天，为便于 getMaxDay 函数的实现，该数组多出一项
    const int DAYS_BEFORE_MONTH[] = { 0, 31, 59, 90, 120, 151, 181, 212, 243,
    273, 304, 334, 365 };
}
Date::Date(int year, int month, int day) : year(year), month(month), day(day) {
    if (day <= 0 || day > getMaxDay()) {
        cout << "Invalid date: ";
        show();
        cout << endl;
        exit(1);
    }
    int years = year - 1;
    totalDays = years * 365 + years / 4 - years / 100 + years / 400
        + DAYS_BEFORE_MONTH[month - 1] + day;
    if (isLeapYear() && month > 2) totalDays++;
}
int Date::getMaxDay() const {
    if (isLeapYear() && month == 2)
        return 29;

```

```

        else
            return DAYS_BEFORE_MONTH[month]-
DAYS_BEFORE_MONTH[month - 1];
    }
    void Date::show() const {
        cout << getYear() << "-" << getMonth() << "-" << getDay();
    }
}

```

```

//account.h
#ifndef __ACCOUNT_H__
#define __ACCOUNT_H__
#include "date.h"
#include <string>
class SavingsAccount { //储蓄账户类
private:
    std::string id;    //帐号
    double balance;    //余额
    double rate;    //存款的年利率
    Date lastDate;    //上次变更余额的时期
    double accumulation; //余额按日累加之和
    static double total; //所有账户的总金额
    //记录一笔帐, date 为日期, amount 为金额, desc 为说明
    void record(const Date &date, double amount, const std::string &desc);
    //报告错误信息
    void error(const std::string &msg) const;
    //获得到指定日期为止的存款金额按日累积值
    double accumulate(const Date& date) const {
        return accumulation + balance * date.distance(lastDate);
    }
public:
    //构造函数
    SavingsAccount(const Date &date, const std::string &id, double rate);
    const std::string &getId() const { return id; }
    double getBalance() const { return balance; }
    double getRate() const { return rate; }
    static double getTotal() { return total; }
    //存入现金
    void deposit(const Date &date, double amount, const std::string &desc);
    //取出现金
    void withdraw(const Date &date, double amount, const std::string &desc);
    //结算利息, 每年 1 月 1 日调用一次该函数
    void settle(const Date &date);
}

```

```

        //显示账户信息
        void show() const;
    };
#endif //__ACCOUNT_H__

//account.cpp
#include "account.h"
#include <cmath>
#include <iostream>
using namespace std;
double SavingsAccount::total = 0;
//SavingsAccount 类相关成员函数的实现
SavingsAccount::SavingsAccount(const Date &date, const string &id, double
rate)
    : id(id), balance(0), rate(rate), lastDate(date), accumulation(0) {
    date.show();
    cout << "\t#" << id << " created" << endl;
}
void SavingsAccount::record(const Date &date, double amount, const string
&desc) {
    accumulation = accumulate(date);
    lastDate = date;
    amount = floor(amount * 100 + 0.5) / 100; //保留小数点后两位
    balance += amount;
    total += amount;
    date.show();
    cout << "\t#" << id << "\t" << amount << "\t" << balance << "\t" << desc
<< endl;
}
void SavingsAccount::error(const string &msg) const {
    cout << "Error(#" << id << "): " << msg << endl;
}
void SavingsAccount::deposit(const Date &date, double amount, const string
&desc) {
    record(date, amount, desc);
}
void SavingsAccount::withdraw(const Date &date, double amount, const string
&desc) {
    if (amount > getBalance())
        error("not enough money");
    else
        record(date, -amount, desc);
}
void SavingsAccount::settle(const Date &date) {

```

```

        double interest = accumulate(date) * rate    //计算年息
                                   / date.distance(Date(date.getYear() - 1, 1, 1));
        if (interest != 0)
            record(date, interest, "interest");
        accumulation = 0;
    }
    void SavingsAccount::show() const {
        cout << id << "\tBalance: " << balance;
    }
}

```

```

//main.cpp
#include "account.h"
#include <iostream>
using namespace std;
int main() {
    Date date(2008, 11, 1);    //起始日期
    //建立几个账户
    SavingsAccount accounts[] = {
        SavingsAccount(date, "03755217", 0.015),
        SavingsAccount(date, "02342342", 0.015)
    };
    const int n = sizeof(accounts) / sizeof(SavingsAccount); //账户总数
    //11 月份的几笔账目
    accounts[0].deposit(Date(2008, 11, 5), 5000, "salary");
    accounts[1].deposit(Date(2008, 11, 25), 10000, "sell stock 0323");
    //12 月份的几笔账目
    accounts[0].deposit(Date(2008, 12, 5), 5500, "salary");
    accounts[1].withdraw(Date(2008, 12, 20), 4000, "buy a laptop");
    //结算所有账户并输出各个账户信息
    cout << endl;
    for (int i = 0; i < n; i++) {
        accounts[i].settle(Date(2009, 1, 1));
        accounts[i].show();
        cout << endl;
    }
    cout << "Total: " << SavingsAccount::getTotal() << endl;
    return 0;
}

```

### 三、 实验要求

- 1、 完成 PTA 作业《C++语言程序设计工程实践：内存编程》的所有

实验题。

- 2、 成功在 Visual Studio 2019 里面运行上述代码。